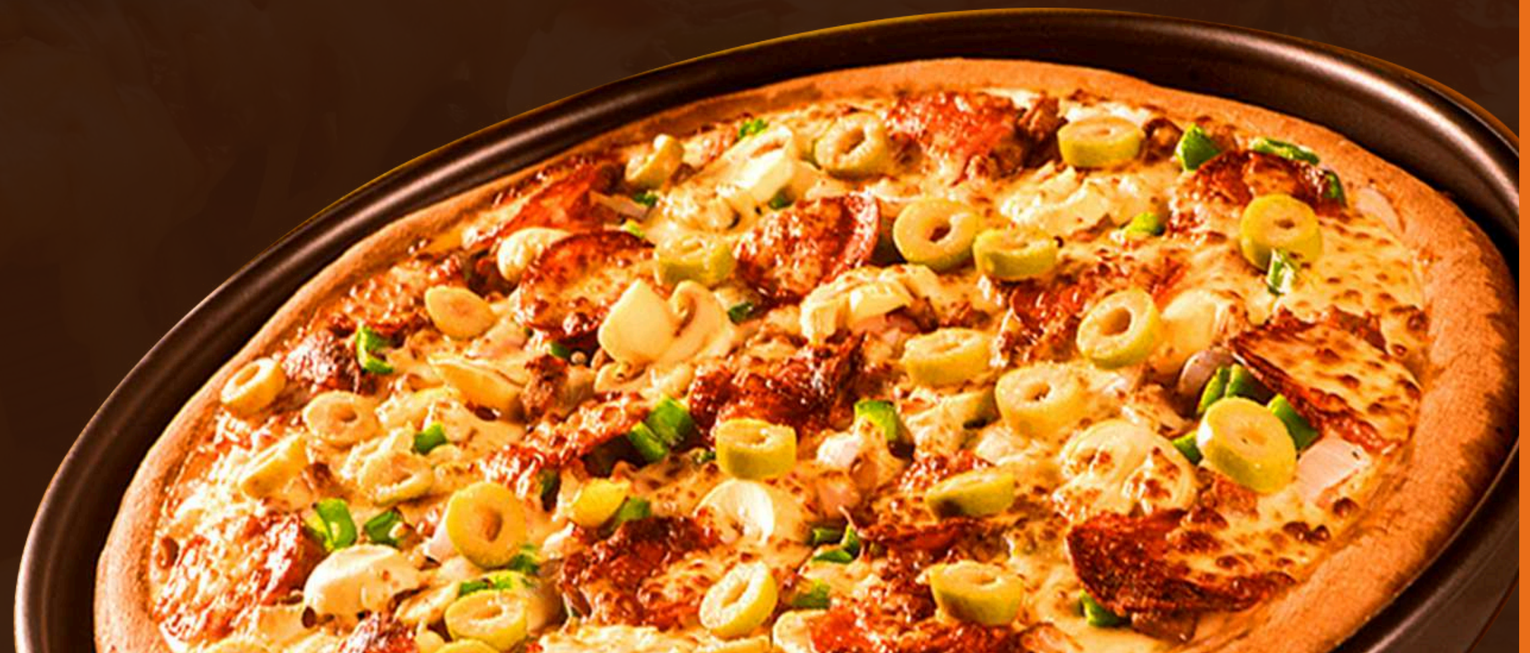
PIZZA SALES ANALYSIS USING SQL



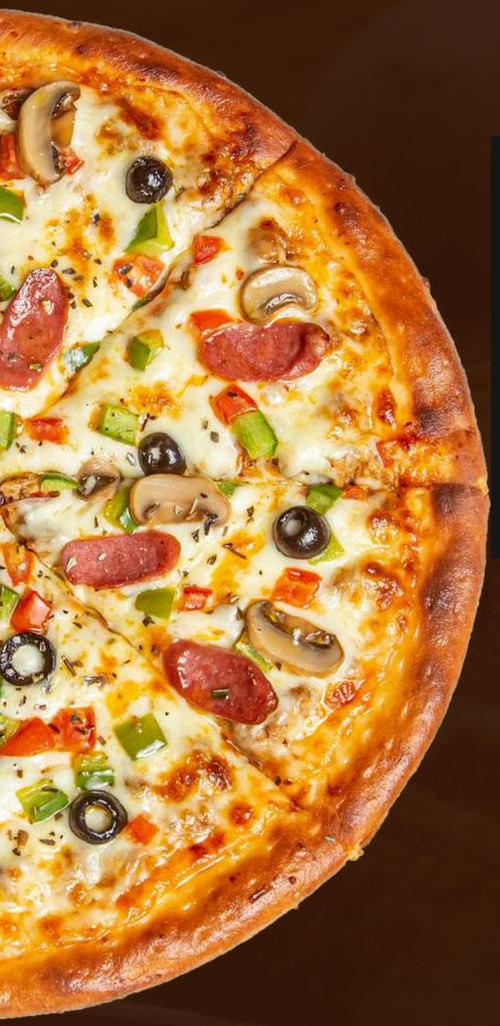
NISHANT KUMAR
PGDM-BUSINESS ANALYTICS

Project Overview

Objective: Enhance decision-making in sales and inventory management through data insights.

- **Utilized SQL to analyze pizza sales data from different tables.**
- **Performed data cleaning, joined tables, and extracted key business insights.**
- **Focused on identifying sales trends, revenue drivers, and optimization strategies.**

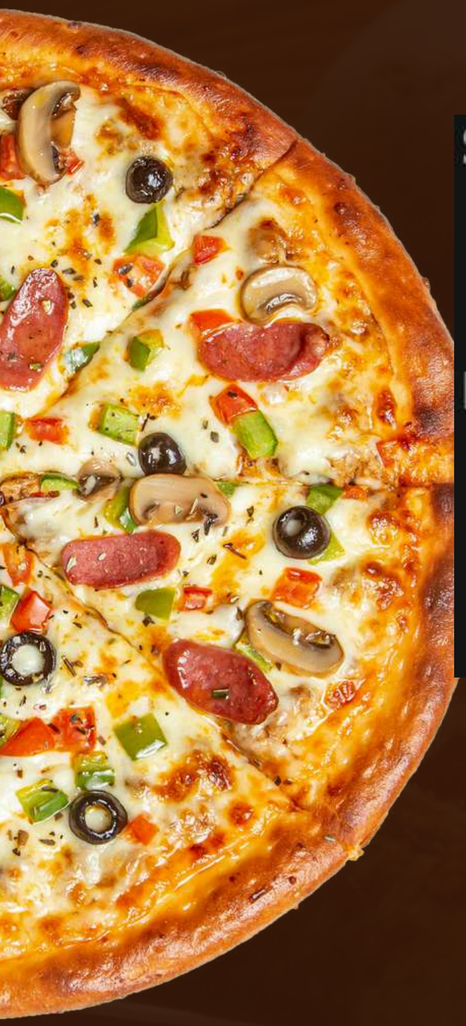
TOTAL ORDERS PLACED



```sql
SELECT
    COUNT(order_id) AS Total_Orders
FROM
    ORDERS;
```

Total_Orders
21350

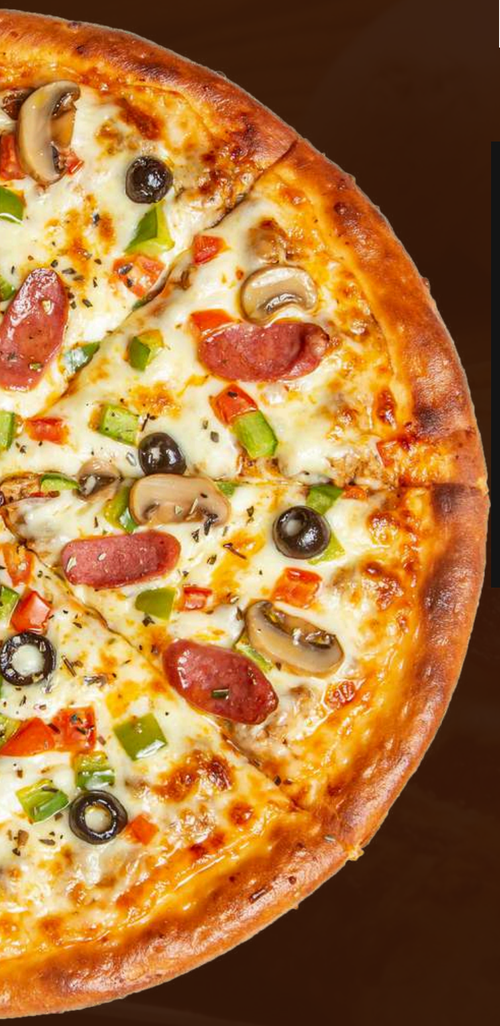
TOTAL REVENUE GENERATED



```
SELECT
    ROUND(SUM(ORDER_details.quantity * pizzas.price),
            2) AS Total_sales
FROM
    ORDER_details
        JOIN
    pizzas ON ORDER_details.pizza_id = pizzas.pizza_id
```

Total_sales
817860

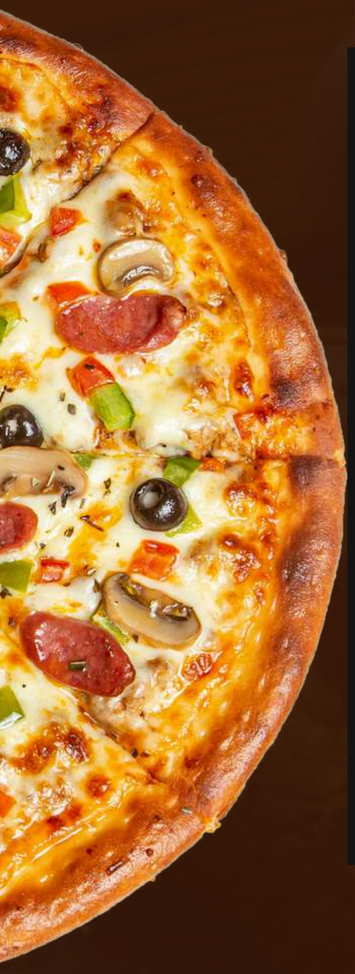
HIGHEST-PRICED PIZZA



```sql
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        INNER JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY price DESC LIMIT 1;
```

name	price
The Greek Pizza	35.95

MOST COMMON PIZZA SIZE ORDERED

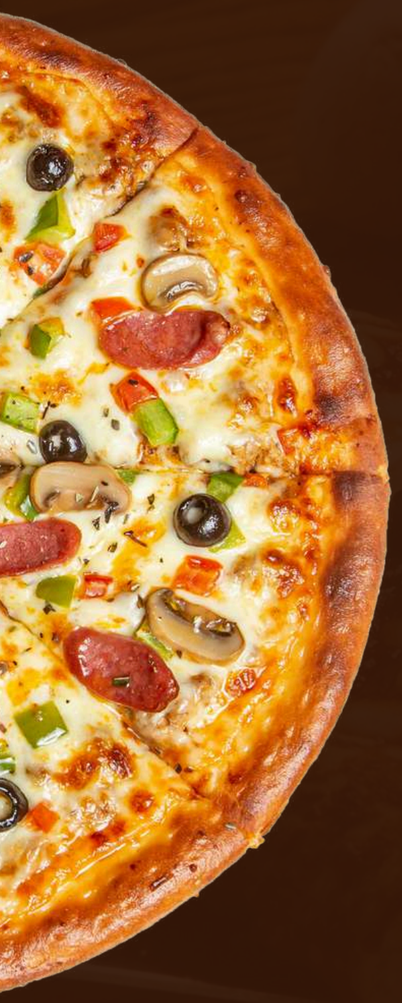


```sql
SELECT
    quantity, COUNT(order_detail_id)
FROM
    ORDER_details
GROUP BY quantity;

SELECT
    pizzas.size,
    COUNT(ORDER_details.order_detail_id) AS order_count
FROM
    pizzas
        JOIN
    ORDER_details ON pizzas.pizza_id = ORDER_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

TOP 5 PIZZA TYPES BY QUANTITY



```sql
SELECT
    pizzas.pizza_type_id,
    SUM(ORDER_details.quantity) AS Total_pizza_type_oders
FROM
    ORDER_details
        INNER JOIN
    pizzas ON ORDER_details.pizza_id = pizzas.pizza_id
GROUP BY pizzas.pizza_type_id
ORDER BY Total_pizza_type_oders DESC
LIMIT 5;
```

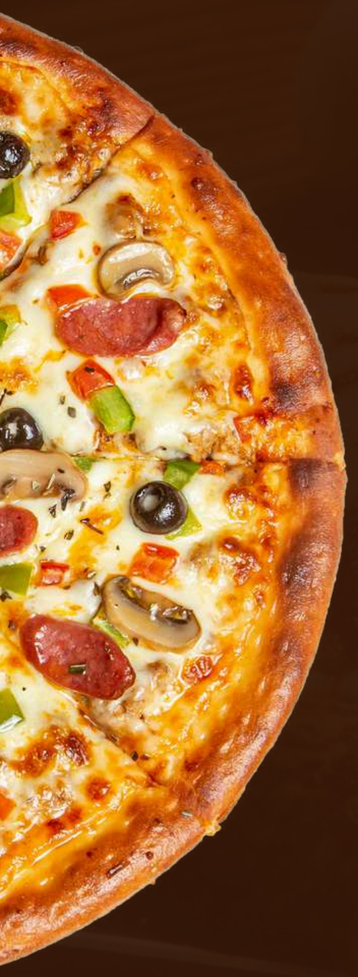
pizza_type_id	Total_pizza_
classic_dlx	2453
bbq_ckn	2432
hawaiian	2422
pepperoni	2418
thai_ckn	2371

TOTAL QUANTITY BY PIZZA CATEGORY

```sql
SELECT
    pizza_types.category,
    SUM(ORDER_details.quantity) AS Quantity
FROM
    pizza_types
        INNER JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    ORDER_details ON ORDER_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY Quantity DESC;
```

category	Quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

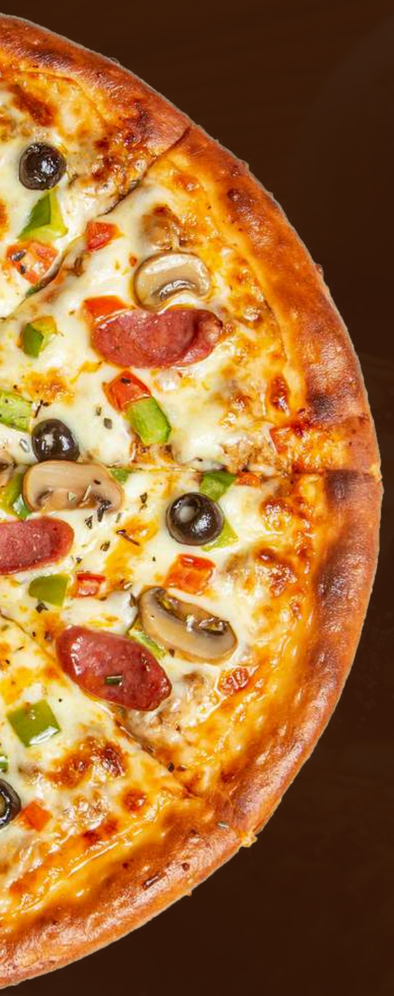
ORDERS DISTRIBUTION BY HOUR



```sql
SELECT
    HOUR(order_time) AS Hour, COUNT(order_id) AS Order_count
FROM
    ORDERS
GROUP BY HOUR(order_time);
```

Hour	Order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

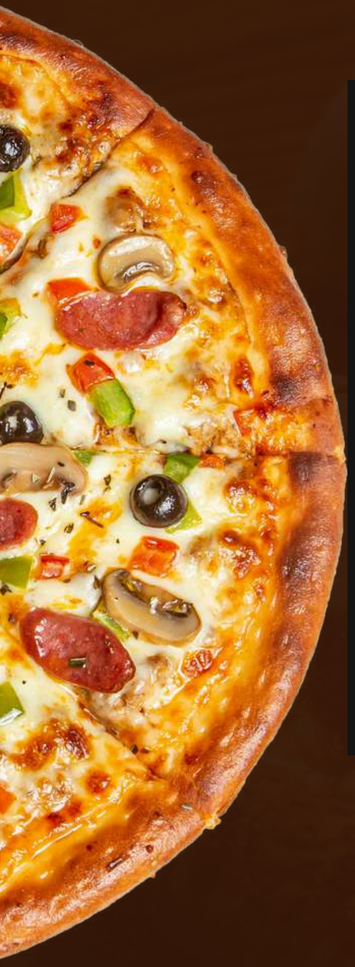
AVERAGE PIZZAS ORDERED PER DAY



```sql
SELECT
    ROUND(AVG(quantity), 0) AS AVERAGE_PIZZAS_ORDERED_PER_DAY
FROM
    (SELECT
        ORDERS.order_date, SUM(ORDER_details.quantity) AS quantity
    FROM
        ORDER_details
    JOIN ORDERS ON ORDER_details.order_id = ORDERS.order_id
    GROUP BY ORDERS.order_date) AS order_quantity;
```

AVERAGE_PIZZAS_ORDERED_PER_DAY
138

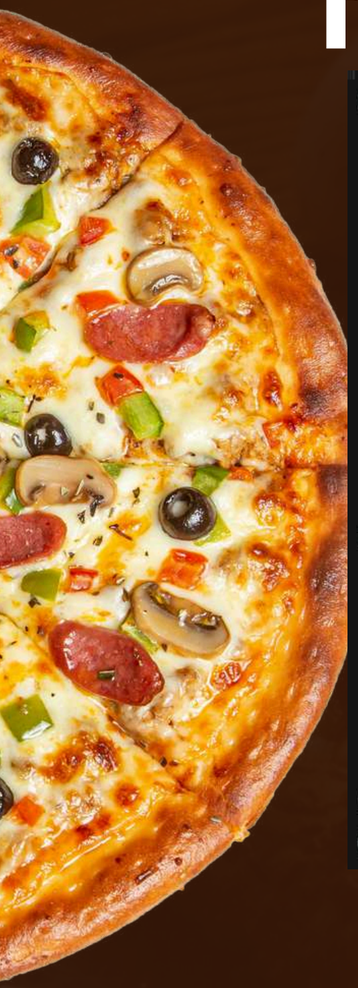
TOP 3 PIZZAS BY REVENUE



```sql
SELECT
    pizza_types.name,
    SUM(ORDER_details.quantity * pizzas.price) AS Revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    ORDER_details ON ORDER_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY Revenue DESC LIMIT 3;
```

name	Revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

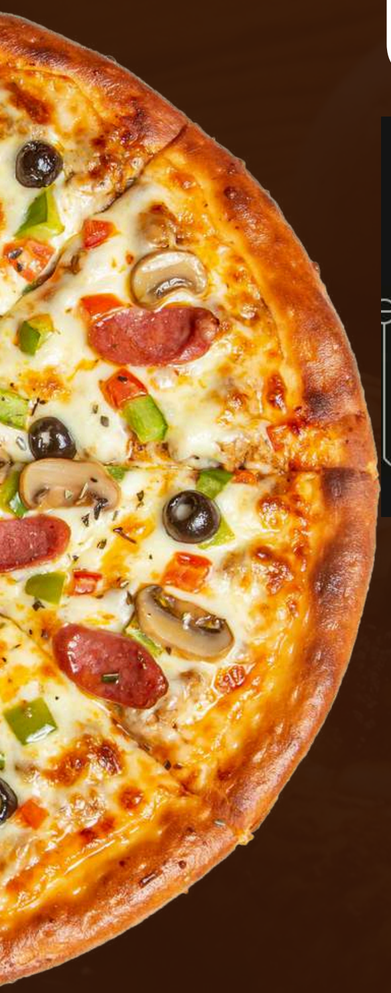
REVENUE CONTRIBUTION BY PIZZA TYPE



```sql
SELECT
    pizza_types.category,
    ROUND(SUM(ORDER_details.quantity * pizzas.price) / (SELECT
                    ROUND(SUM(ORDER_details.quantity * pizzas.price),
                                2) AS Total_sales
                FROM
                    ORDER_details
                        JOIN
                    pizzas ON ORDER_details.pizza_id = pizzas.pizza_id) * 100,
            2) AS REVENUE
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    ORDER_details ON ORDER_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY Revenue DESC;
```

category	REVENUE
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

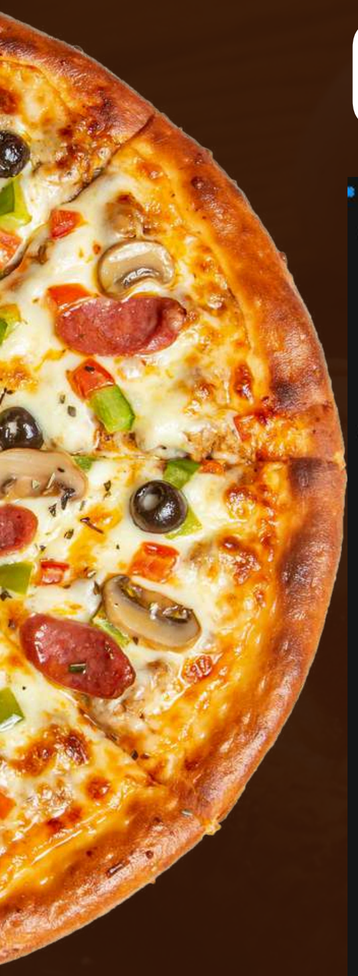
CUMULATIVE REVENUE OVER TIME



```
SELECT order_date ,
SUM(revenue) OVER(order BY Order_date) AS CUM_revenue
FROM
(SELECT ORDERS.order_date , SUM(ORDER_details.quantity*pizzas.price) AS Revenue
FROM ORDER_details JOIN  pizzas ON ORDER_details.pizza_id = pizzas.pizza_id
JOIN ORDERS ON ORDER_details.order_id=ORDERS.order_id
GROUP BY ORDERS.order_date) AS sales ;
```

order_date	CUM_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001
2015-01-17	39001.75000000001
2015-01-18	40978.600000000006
2015-01-19	43365.75000000001
2015-01-20	45763.65000000001
2015-01-21	47804.20000000001
2015-01-22	50300.90000000001
2015-01-23	52724.600000000006
2015-01-24	55013.850000000006
2015-01-25	56631.40000000001
2015-01-26	58515.80000000001

TOP 3 PIZZA TYPES BY REVENUE WITHIN EACH CATEGORY

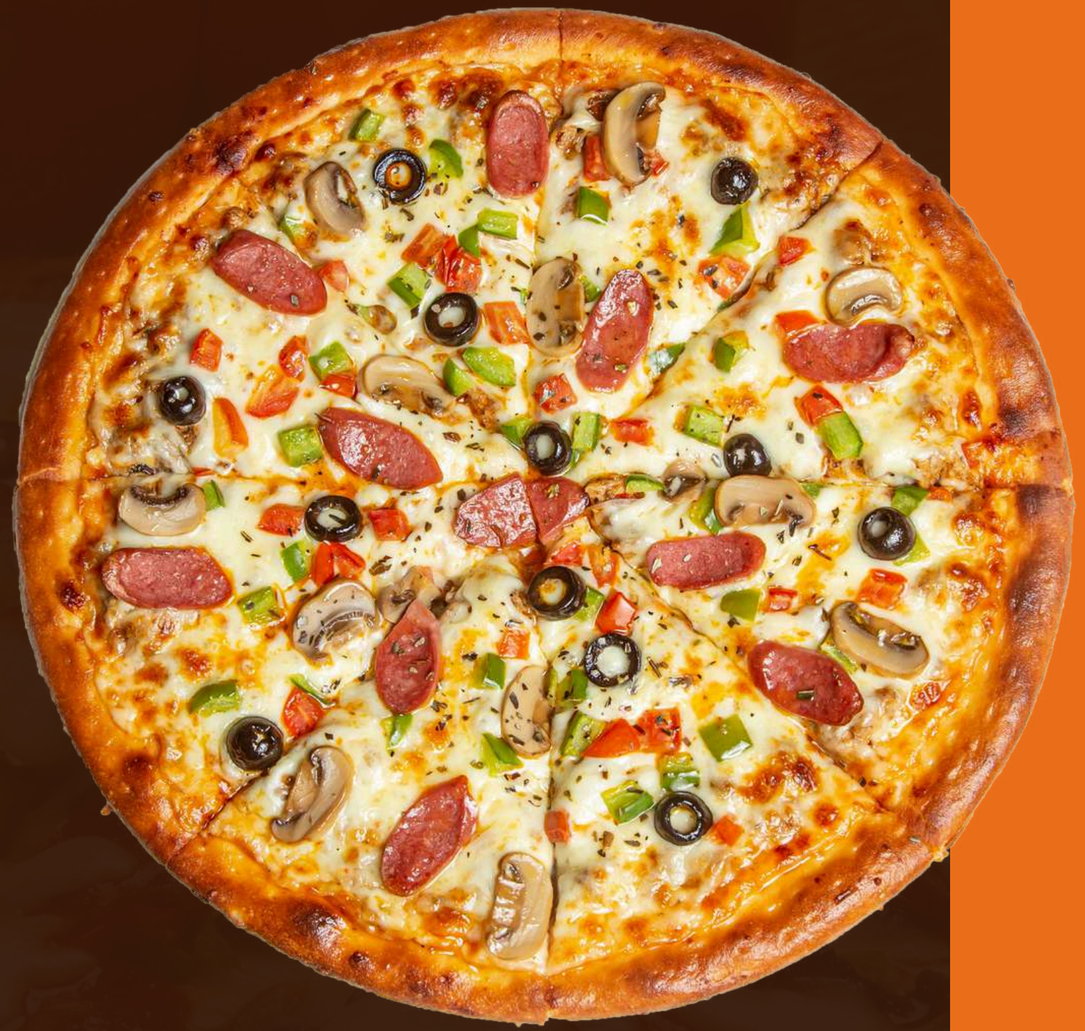


```sql
SELECT name,category,Revenue FROM
(SELECT
    category,
    name,
    Revenue,
    RANK() OVER(PARTITION BY category ORDER BY Revenue DESC) AS Rn
FROM (
    SELECT
        pt.category,
        pt.name,
        SUM(od.quantity * p.price) AS Revenue
    FROM
        pizza_types pt
    JOIN
        pizzas p ON pt.pizza_type_id = p.pizza_type_id
    JOIN
        ORDER_details od ON od.pizza_id = p.pizza_id
    GROUP BY
        pt.category, pt.name
) AS RankedData) AS b
WHERE Rn<=3;
```

name	category	Revenue
The Thai Chicken Pizza	Chicken	43434.25
The Barbecue Chicken Pizza	Chicken	42768
The California Chicken Pizza	Chicken	41409.5
The Classic Deluxe Pizza	Classic	38180.5
The Hawaiian Pizza	Classic	32273.25
The Pepperoni Pizza	Classic	30161.75
The Spicy Italian Pizza	Supreme	34831.25
The Italian Supreme Pizza	Supreme	33476.75
The Sicilian Pizza	Supreme	30940.5
The Four Cheese Pizza	Veggie	32265.700(
The Mexicana Pizza	Veggie	26780.75
The Five Cheese Pizza	Veggie	26066.5

RECOMMENDATIONS

- Inventory Management: Adjust based on demand trends identified.
- Peak Hour Promotions: Special offers during peak hours to boost sales.
- Sales Strategy: Focus on high-revenue pizza types for increased profitability.
- Dynamic Pricing: Experiment with pricing strategies for less popular pizzas.



THANK YOU